



**QUEEN'S  
UNIVERSITY  
BELFAST**

## **DRAM Characterization under Relaxed Refresh Period Considering System Level Effects within a Commodity Server**

Mukhanov, L., Tovletoglou, K., Nikolopoulos, D., & Karakonstantis, G. (2018). DRAM Characterization under Relaxed Refresh Period Considering System Level Effects within a Commodity Server. In *2018 IEEE 24th International Symposium on On-Line Testing and Robust System Design, IOLTS 2018* (pp. 236- 239). [8474184] (2018 IEEE 24th International Symposium on On-Line Testing and Robust System Design, IOLTS 2018). Institute of Electrical and Electronics Engineers Inc.. <https://doi.org/10.1109/IOLTS.2018.8474184>

**Published in:**

2018 IEEE 24th International Symposium on On-Line Testing and Robust System Design, IOLTS 2018

**Document Version:**

Peer reviewed version

**Queen's University Belfast - Research Portal:**

[Link to publication record in Queen's University Belfast Research Portal](#)

**Publisher rights**

© 2018 IEEE. This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

**General rights**

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact [openaccess@qub.ac.uk](mailto:openaccess@qub.ac.uk).

# DRAM Characterization under Relaxed Refresh Period Considering System Level Effects within a Commodity Server

**Abstract**—Today’s rapid generation of data and the increased need for higher memory capacity has triggered a lot of studies on aggressive scaling of refresh period, which is currently set according to rare worst case conditions. Such studies analysed in detail the data-dependent circuit level factors and indicated the need for online DRAM characterization due to the variable cell retention time. They have done so by executing few test data patterns on FPGAs under controlled temperatures by using thermal testbeds, which however cannot be available in the field. Moreover, the existing studies were not able to reveal the system level effects, which may be excited under the execution of workloads on real systems and directly or indirectly affect DRAM reliability. In this paper, we develop a first of its kind experimental framework based on a state-of-the-art 64-bit ARM based server with Linux OS, in which we enabled the DRAM characterization under relaxed refresh period by executing conventional test data patterns as well as popular HPC and Cloud workloads. Such a setup allows us for the first time to evaluate the impact of any system level factors on DRAM behaviour and the efficacy of conventional test patterns in typical conditions without controlling the DRAM temperature. Our results indicate that common test patterns are ineffective in identifying error-prone locations at low DRAM temperatures. Furthermore, the analysis of various measured performance counters and manifested error rates reveal that there is a strong correlation between system utilization and DRAM reliability. By exploiting such findings, we developed a benchmark, which can indirectly stress the DRAM temperature and thus used for characterization in the field without needing any complicated thermal equipment. Results show that the stress benchmark can increase the DRAM temperature above  $43^{\circ}\text{C}$  and cover up to 60% of erroneous memory locations in the 144 tested DRAM chips. Finally, our study shows for the first time that the refresh period can be relaxed by 35 times on such a commodity system with all errors being corrected by the available error correcting codes, resulting in 11.5% power savings on average.

## I. INTRODUCTION

The rapid growth of connected Internet-of-Things (IoT) is estimated to generate by 2020 24.3 exabytes of data [1] creating immense needs for more data storage capacity and aggressive scaling of DRAMs. Such needs have already turned DRAM based subsystems into one of the main power consumers, especially in servers, with estimations indicating that soon they will be accountable for almost half of the consumed power [2]. Many studies tried to address this reality by reducing the refresh power, estimated to incur 40 % overhead in future 64Gb densities due to the conservative selection of the refresh period needed for addressing the limited retention time of the DRAM cells [3]. The majority of the proposed schemes rely on offline identification of the weak cells using few known data patterns and the adoption of different refresh periods for various cells, rows and pages [3], [4], [5], [6], [7], [8]. However, recent studies have proven such schemes ineffective, since they revealed that the cell retention time varies dynamically due to data-dependent circuit level crosstalk effects [3], [9]. Such effects are even worse under the extreme temperatures that are applied during the DRAM characterization, which was one more reason for urging the need for online DRAM characterization in the field [9].

However, it is questionable if the conventionally used data patterns will be effective for characterizing the DRAMs within servers in the field, where any thermal testbed (commonly used to stress the DRAM temperature in existing studies) will be unavailable. Furthermore, existing characterization campaigns were performed on FPGAs under fixed DRAM temperatures [9], [10], [11], [12], not allowing them to

study any dynamic system level effects which may be excited by any executed application within a server and directly or indirectly affect DRAM reliability. Such an impact on DRAM reliability, caused indirectly by the system utilization, was also suspected in a long term study in a Google data center [13] but have never been thoroughly studied. Investigating such effects will require the execution of applications on a server with non-controlled DRAM temperature, while collecting and analysing relevant performance counters, which have not been done by current FPGA based campaigns.

This paper attempts to address such challenges by making the following contributions:

- We develop a novel experimental framework for characterizing DRAMs under relaxed refresh period, within a state-of-the-art 64-bit ARM based server. Such a setup allows us to study for the first time the impact of potential system level factors that may be excited by applications within a commodity server.
- We perform a first of its kind characterization of the power-reliability trade-off of 144 server grade DRAM chips under scaled refresh period using conventional data patterns as well as High-Performance Computing (HPC) and Cloud workloads. Our results reveal that the identified erroneous locations and the DRAM temperature vary across applications, which has not been considered in the previous FPGA studies, where the DRAM temperature was fixed during experiments. We also observe that the conventional data patterns are ineffective in revealing error-prone locations without any thermal assist mechanism even after relaxing refresh period by 35x.
- To quantify the impact of system level factors, we collect various performance counters and correlate them with the failure rates, revealing that the System-on-Chip (SOC) utilization may indirectly affect the DRAM temperature and thus its reliability.
- We design a micro-benchmark that combines the random data pattern and higher SOC utilization to indirectly increase the DRAM temperature, providing an effective mechanism for online DRAM characterization without any thermal testbed. Results show that such a benchmark can increase the DRAM temperature up to  $43^{\circ}\text{C}$  and identify most error-prone locations.
- We demonstrate for the first time that the total memory power could be reduced by 11.5 % on average in a such server with a complete software stack by relaxing the refresh period by 35x. This was possible without compromising the system availability since Single-Error-Correction Double-Error-Detection Error Correction Code (SEDED ECC) was adequate for correcting all manifested errors and avoiding any system crashes.

The rest of the paper is organized as follows. Section II describes the background and challenges, while Section III presents our experimental framework. Section IV analyses the results of our experimental campaign. Finally, conclusions are drawn in Section V.

## II. DRAM BACKGROUND AND CHALLENGES

A main memory sub-system based on DRAMs is organized hierarchically into channels supporting a number of modules. Each Dual

In-line Memory Module (DIMM) usually has two ranks, each of which consists of DRAM chips. Within each chip, DRAM cells are organized into banks, which are two-dimensional arrays, addressed based on rows and columns.

The main drawback of the DRAM technologies is the limited retention time [10] of the cell's charge. To avoid any error induced by the charge leakage over time, DRAM employs an *Auto-Refresh* mechanism that periodically recharges all cells in the array. Conventionally, all DDR technologies adopt today a *refresh period*,  $T_{REFP}$ , of 64 ms for refreshing periodically each cell of the DIMM based on the worst case retention time across all cells. However, in reality many cells have a much higher retention time and the operating conditions may not be as bad as the ones assumed [12], [14]. It was shown that such a pessimistic  $T_{REFP}$  leads to considerable power and performance overheads, which are expected to worsen as the DRAM density increases [10].

#### A. State-of-the-Art and Challenges

This reality has urged many researchers to study the DRAM behaviour under scaled  $T_{REFP}$  and suggest methods to relax the refresh operations [15], [3], [16]. The majority of existing schemes rely on offline DRAM cell characterization, either to group cells according to their retention time and adjust the  $T_{REFP}$  for each group separately [3], [5], [6] or to design tailored error mitigation strategies [15], [16], [7].

Typically, existing schemes characterize the retention time of cells using a set of data patterns executed on FPGA based setups. Such studies perform experiments under various DRAM temperatures, which are controlled using thermal testbeds, since it was shown that the retention time of DRAM cells decreases exponentially as temperature increases [17], [10], [12]. Using such setups, recent findings showed that trying to profile the retention time of all weak cells offline, before the deployment of the DRAM chips, is extremely challenging as the same cell can fail or operate correctly at different times independent of temperature due to data-dependent circuit level crosstalk effects [10], [9]. Manufacturers spend even days on testing each single DRAM chip for identifying cells prone to such effects, and it was estimated that the testing time and cost will further increase as retention failures become more prominent with technology scaling [9]. Such findings motivated the need for online DRAM profiling after deployment of the DRAM chips.

All existing studies may have exploited the spatial distribution of retention time and revealed the impact of various circuit level data- and temperature- dependent effects, however they still left a number of questions unanswered. First of all, existing studies were performed on custom FPGA setups, which may help simplify the characterization process, however they do not allow to study the impact of system level effects, which may be excited within a server and affect DRAM reliability. Typically, DIMMs are being placed in slots on a single motherboard adjacent to the SOC which has integrated the Main Processing Modules (PMDs) and the Memory Controller Units (MCUs), as shown in Figure 1, which depicts the system layout of a state-of-art 64-bit ARM based server. In modern processors, the northbridge (or memory controller), which controls and transfers data to/from DIMMs, is integrated with the chip to reduce memory access latencies. However, these latencies also depend on the physical distance between the SOC and DIMMs [18] which makes manufactures place memory slots on the sides of processor sockets in many server grade motherboards like the standard ATX [19].

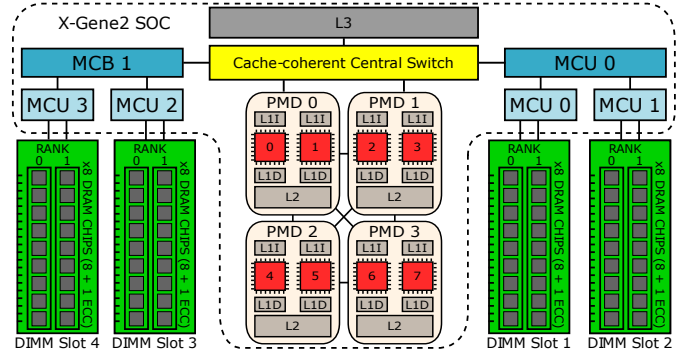


Fig. 1: Block diagram of the X-Gene2 SOC.

DRAMs are essentially part of deep memory hierarchies including multi level caches. Several parameters of such hierarchies in real systems, like the organization, the size of the caches and the supported memory bandwidth, could directly affect the accesses to DRAM and thus its reliability. Previous long term studies on the DRAM behaviour in Google data centers [13] have suggested that system utilization within a server could also indirectly influence the DRAM temperature and thus its reliability profile. The thorough study of such effects requires the measurement of various performance counters from each server, which was not performed by this study and are not applicable for FPGAs.

Furthermore, to allow the investigation of any direct or indirect system level effects and of system utilization on DRAM reliability, experiments need to be performed under non-controlled DRAM temperature as opposed to existing studies which control the DRAM temperature to fixed values. During these experiments, various performance counters, such as Instructions Per Clock (IPC), SOC utilization, number of cache and memory accesses, need to be considered and correlated to the manifested errors.

What is more, real workloads are expected to dynamically change the system load, which is reflected by various performance counters, as well as the SOC and DRAM temperatures and thus may cause different system level effects that may influence dynamically the DRAM behaviour. It would be also interesting to compare such behaviour to the one observed by executing conventional data patterns under non-controlled temperature to evaluate their efficacy in discovering error-prone locations in the field without any thermal stress testbeds.

Finally, there is a need to investigate if the available ECC on server grade DRAMs is sufficient in correcting all manifested errors, while evaluating the power savings under relaxed refresh periods.

### III. EXPERIMENTAL CHARACTERIZATION FRAMEWORK

To address the challenges raised in Section II, we develop a systematic experimental framework, which we will describe in detail.

#### A. Infrastructure Details

The basis of our experimental framework is a state-of-the-art commodity 64-bit ARMv8 based server, the X-Gene2 Server-on-a-Chip, which is the latest generation of the X-Gene family of chips used in the popular HP Moonshot servers [20]. As depicted on Figure 1, the X-Gene2 SOC consists of four PMDs, each with two 64-bit ARMv8 cores running at 2.4GHz. The implemented memory hierarchy is representative of any modern high performance system consisting of a 32 KB L1 data cache and a 32 KB L1 instruction cache per core, a private 256 KB L2 cache shared between the two cores of each PMD and an 8 MB L3 cache shared across all four PMDs through the cache-coherent Central Switch (CSW). The X-Gene2 has two Memory Controller Bridges (MCBs) which are connected to the

CSW providing access to DRAM. In turn, each MCB is connected to two DDR3 MCUs. Each MCU has one channel of DDR3 memory and support up to two DIMMs with two ranks each. In our campaign, we are experimenting independently with two different sets of 4 Micron DDR3 8GB DIMMs at 1866 MHz [21], one DIMM per MCU. In total, we are characterizing 144 chips of 4Gb x8 DDR3 [22], since each DIMM includes 16 and 2 DRAM chips for data storage and ECC, respectively.

The X-Gene2 provides access to a separate Scalable Lightweight Intelligent Management Processor (SLIMpro), a special management core, which is used to boot the system and provide access to on board sensors for measuring the temperature and power of the SOC and DRAM. The SLIMpro also reports to the Linux kernel all memory errors corrected or detected by SECDED ECC, providing information about the DIMM, bank, rank, row and column that the error occurred. The available ECC can detect and correct single-bit errors in a 64-bit word, which we refer to as *correctable errors* (CEs) and detect two-bit errors that cannot be corrected, which we refer to as *uncorrectable errors* (UEs). Finally, SLIMpro allows to configure the parameters of the MCUs, such as timings and  $T_{REFP}$ . The server runs a fully-fledged OS based on CentOS 7 with the default Linux kernel 4.3.0 for ARMv8 and support for 64KB pages.

### B. Characterization Benchmarks

For our characterization campaign, we selected a set of data patterns micro-benchmarks (DPBenchs) which were used in all previous retention characterization studies [10], [23]. In particular, we used three DPBenchs with static data patterns: one with *all0s*, one with *all1s*, and one with a *checkerboard* pattern as well as a dynamic data pattern based on *random* data (uniformly distributed). Apart from the DPBenchs, we also used a set of server workloads to satisfy one of the primary aims of this work about studying the system level effects. In particular, we used Speckle Reducing Anisotropic Diffusion (*srad*) and NeedlemanWunsch (*nw*) kernels from the Rodinia HPC Benchmark Suite [24]. In our experiments, we run single-threaded (*srad*(1), *nw*(1)) and eight-threaded (*srad*(8), *nw*(8)) versions of the benchmarks to evaluate how parallel access patterns affect memory reliability. Furthermore, we selected to use popular Cloud workloads from the CloudSuite [25]. In particular, we use a Distributed Memory Caching *memcached*, a Graph Analytics *graph-analytics* and a Web Search *web-search* workload deployed with Docker. Note that all these benchmarks and especially the CloudSuite workloads are being ported and characterized on a 64-bit ARM server for the first time.

### C. DRAM Characterization Flow

As we said the primary target of our campaign is to characterize the DRAM within a commodity server, therefore in our experiments we relax the  $T_{REFP}$  from 64 *ms* to 2.283 *s* that is the maximum allowed  $T_{REFP}$  in the X-Gene2 server, while keeping the supply voltage to the default 1.5V. Note that in our experiments, the ARM cores operate at the default frequency of 2.4GHz.

The server is placed in a rack within a room with controlled ambient temperature of 18 °C, such conditions are representative within any data center, which may vary from 15 °C to 22 °C [14]. Note that the average SOC and DRAM temperatures, when the system is at idle are 36 °C and 21 °C, respectively.

**Error Accounting.** We used the aforementioned error reporting mechanisms to record the CE and UE manifested within each 64-bit word under relaxed  $T_{REFP}$ . In addition, to account for any potential errors of more than two-bits in a 64-bit word that cannot be detected

by ECC, we compared the output of each execution with a golden reference output obtained when DRAM is operating at the nominal  $T_{REFP}$ . In this way, essentially we were able to measure any Silent Data Corruption (SDC) that could go undetected by SECDED ECC.

We run each DPBench for a number of *rounds* similar to [10]. Each round consists of three phases: i) writing the data pattern to the allocated memory, ii) waiting for some specific time to make sure that all the cells have been idle for  $T_{REFP}$  and iii) reading back the data for enabling the ECC to detect any errors due to failing cells. We run each DPBench for two hours to ensure that each benchmark makes at least 16 rounds as in [10] which was found that it is adequate for covering most of error-prone locations. Since we are interested in comparing the DRAM behaviour, under the execution of the DPBenchs and the real workloads, we run also each HPC and Cloud workload for two hours which is enough for allowing each of them to execute few iterations and issue a number of memory accesses, representative of real life executions.

During our experiments, we allocated 28 GB memory for each DPBench, the maximum available to user space. In case of the HPC and Cloud workloads, we chose to configure all of them with a memory allocation of 8 GB to enable a fair comparison of any effects and of any number of triggered errors between them. Note that in any case, such workloads when executed in real life may not be using the whole user space.

Since one of our targets is to investigate the system level effects during the execution of workloads, we collected various program metrics using the *perf* tool, such as L1/L2/memory accesses per clock, IPC and the SOC utilization estimated as:  $SOC_{util} = \frac{\sum_{i=1}^N T_{i_{thread}}}{T_{program}}$ , where  $T_{i_{thread}}$  corresponds to time spent by the thread *i* on executing a program, *N* is the number of threads and  $T_{program}$  is the real elapsed time of the program. During our experiments, we also recorded the SOC and DRAM temperatures and power by reading the on board sensors every second.

### D. Analysis Phase

At the end of the experimental campaign, we analyse the results and quantify the DRAM reliability profile using a set of metrics. First of all, we calculate the percentage of error-prone locations discovered when running a DPBench or workload over the total number of error-prone locations detected by all benchmarks as:  $PDEL = \frac{Num_{locations}^X}{\sum_{i=bench} Num_{locations}^i}$ , where  $Num_{locations}^X$  is the number of unique error-prone locations, i.e. in terms of 64-bit words discovered when running the benchmark *X*. This metric evaluates the efficacy of a benchmark in discovering error-prone locations.

To validate and compare our results with error rates observed in previous research studies, we also calculate the fraction of failing 64-bit words for a specific benchmark using:  $FFW = \frac{Num_{locations}^X}{size_{memory}^X}$ , where  $size_{memory}^X$  is the size of memory measured in 64-bit words allocated by the benchmark *X*.

To evaluate the probability of discovering error-prone location across time achieved by each workload and essentially quantify how *PDEL* changes in time, we use:  $PDEL_T(t) = \frac{Num_{locations}^X(t)}{size_{memory}^X}$ , where  $Num_{locations}^X(t)$  is the number of unique error-prone locations discovered for the last *t* minutes (we use 10 minutes in our study) of the experiment with the benchmark *X*.

Finally, we use the Spearman's rank correlation coefficient [26] to formally identify and quantify any dependency between the aforementioned performance metrics and memory errors. This coefficient reflects the monotonic relationship between two variables. This type

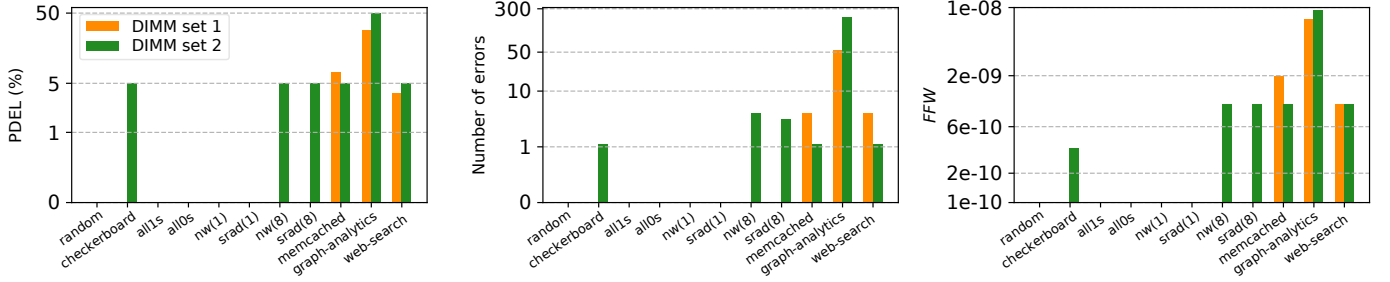


Fig. 2:  $PDEL$ , the total number of errors and  $FFW$  reported across the benchmarks for memory operating under relaxed  $T_{REFP}$ .

of correlation has two output parameters: the correlation coefficient  $r_s$ , which denotes the strength and direction of the correlation, and  $p$ -value for a hypothesis test whose null hypothesis ( $H_0$ ) is that two sets of data are not correlated. The correlation coefficient  $r_s$  is estimated as:  $r_s = 1 - \frac{6 \times \sum d_i^2}{N \times (N^2 - 1)}$ , where  $d_i$  is the difference between two ranks [26] of each observation and  $N$  is the number of samples.

#### IV. CHARACTERIZATION RESULTS

In this section, we present the results of our 3 month experimental campaign focusing on the ones obtained under relaxing the  $T_{REFP}$  to the maximum allowed on the server, i.e. 2.283 s.

##### A. Experiments with Benchmarks under non-controlled temperature

Initially, we present the total number of errors,  $PDEL$  and  $FFW$  obtained after running the DPBenchs, HPC and Cloud workloads without controlling the DRAM temperature to a fixed value and compare our results to the ones obtained by previous research studies.

**DPBenchs:** Figure 2 depicts how  $PDEL$ , the total number of errors and  $FFW$  vary across the DPBenchs for the two DIMM sets. Surprisingly, the  $PDEL$  is zero in case of *random*, *all1s* and *all0s* DPBenchs since they do not essentially manifest any errors on neither of the two DIMM sets. In case of *checkerboard*, the  $PDEL$  is also very small since it triggered only one CE during all our experiments. Note also that we have not discovered any UEs or SDCs in experiments with the DPBenchs.

**HPC and Cloud Workloads:** Similarly to our experiments with the DPBenchs, we discovered only CEs and no UEs or SDCs when we executed the considered HPC and Cloud workloads. At the same time, we see in Figure 2 that HPC and Cloud workloads trigger many more errors than the DPBenchs. Moreover, it is observed that the same workloads may manifest different errors, i.e. location and number, on separate DIMM sets. For example, *nw(8)* and *srad(8)* trigger errors only on the second DIMM set, which can be attributed to manufacturing process variations. Meanwhile, the *graph-analytics* benchmark has the highest  $PDEL$ , the highest total number of reported errors and the highest  $FFW$  among all benchmarks on both DIMM sets, which implies a certain dependence between a running application and the DRAM error behaviour.

##### B. DRAM Temperature Variation across Benchmarks

To further investigate these results, we measure the temperature per DIMM slot (one DIMM per MCU), by using the temperature sensor on the SPD chip [27] of each DIMM across all benchmarks. Figure 3 shows the DRAM temperature per memory slot averaged over the experiments with two DIMM sets with 95 % confidence intervals. We observe that the DRAM temperature averaged over DIMMs varies from 33 °C up to 43 °C.

To understand the results of characterization presented above, we need to keep in mind that these temperatures are less than

the ones (from 45 °C to 85 °C) used by existing studies with thermal testbeds [10], [11], [9], [12]. According to these studies, the fraction of cells with retention time less than 2 s may vary from  $10^{-9}$  up to  $10^{-6}$ . If we impose an assumption that there is only single-bit errors within a 64-bit word, then we should observe from  $10^{-9} \times 28GB \times \frac{8bits}{8bytes} = 240$  up to  $10^{-6} \times 28GB \times \frac{8bits}{8bytes} = 240518$  unique erroneous locations for the DPBenchs and from 69 up to 68719 different error-prone locations for the HPC and Cloud workloads allocating 8 GB of memory. However, in our experiments the DPBenchs triggered only one CE which is by far away from the bottom boundary. We explain this by the low DRAM temperature incurred by the DPBenchs (33 °C on average), except of *checkerboard* which manifested one CE. These findings indicate that if such conventional DPBenchs are used to characterize the DRAMs online, without any mechanism to elevate the temperature, then they will be ineffective.

The highest  $FFW(9.31 \times 10^{-9})$  is observed for *graph-analytics* which induces errors in 28 different memory locations. Nonetheless, this number is still less than the bottom boundary of the range ([69, 68719]) estimated for the HPC and Cloud workloads. We also explain this by the temperature factor: the average DRAM temperature incurred by *graph-analytics* is 41.8 °C which is still lower than the temperature range used in FPGA based studies.

In our study, we run experiments by placing the available DIMMs on different slots. We found that DIMMs placed on the 1st memory slot have the highest temperature, which explains the fact that the majority of errors were reported for this slot. Figure 4a shows the spatial and density distribution of the errors between memory slots and memory ranks aggregated over two DIMM sets when we run the Cloud workloads. We present the distribution as a polar plot where  $\Theta$  - axis specifies DIMM slot and rank, while  $\rho$  - axis reflects the number of errors. We see that the highest number of errors occurred in the DIMM from the 1st slot, which we found to be the same for the HPC benchmarks. However, the question arises why the temperature varies between benchmarks so much. What is more, we see that DIMMs in the 1st slot has the highest temperature for all runs (see Figure 3). Yet all memory accesses should be equally distributed

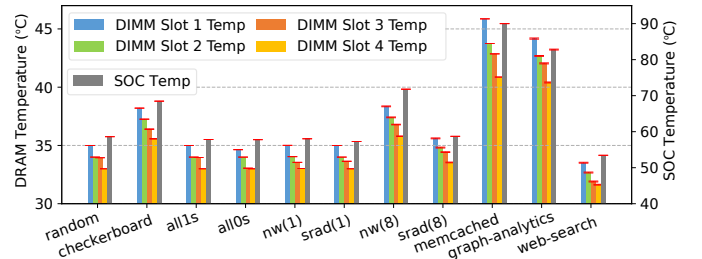


Fig. 3: Average temperatures of each DIMM and of the SOC measured for each benchmark.



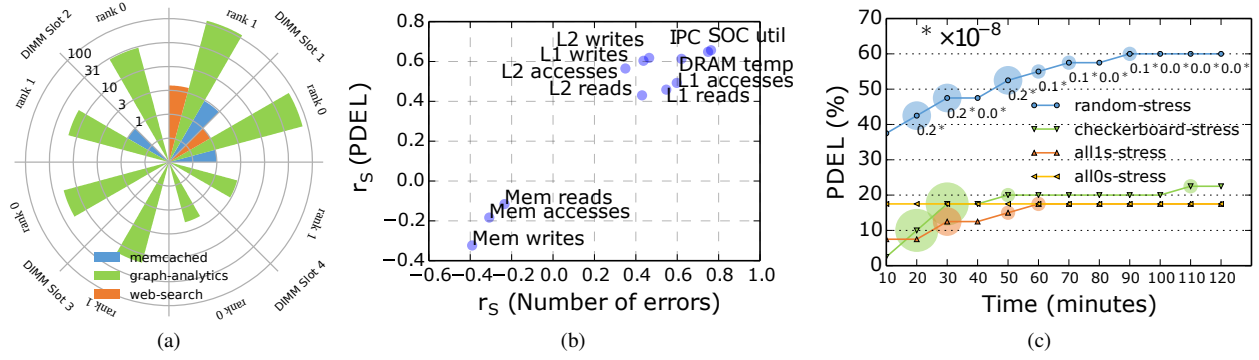


Fig. 4: (a)Distribution of errors between DIMMs slots and ranks for the Cloud workloads, (b)The correlation coefficient  $r_s$  of  $PDEL$  and the total number of CE with a set of system factors, (c) $PDEL$  (Y-axis) and  $PDEL T(10 \text{ minutes})$  for the DPBenchs running with SBench

between DIMMs due to the implemented interleaving mechanism and thus we expect that the temperature of all DIMMs to be similar.

### C. SOC impact on DRAM temperature

We suggest that the temperature of DIMMs is highly correlated with the SOC temperature, which can be seen on Figure 3, the correlation coefficient  $r_s$  is 0.74 ( $\rho$ -value is 0.008). We observe that if the SOC temperature rises then the temperature of all DIMMs also increases, since all memory slots are placed near the chip. Figure 5 displays how slots are placed on the X-Gen2 board and a corresponding thermal photograph of the board. We see that the 1st and 3rd slots are closer to the chip than other slots. Thus, their temperatures should be higher than those of the 2nd and 4th slots. However, our study shows that DIMMs from the 2nd slot often have a higher temperature than the DIMMs placed on the 3rd slot which can be attributed to the air flow or the proximity to other heat sources.

### D. Correlation between System level factors and DRAM reliability

To quantify the effects of various system level factors, including the SOC utilization, on DRAM error behaviour, we collect information about performance and memory access characteristics for each workload (see Section III) and correlate them with  $PDEL$  and the total number of errors across all benchmarks.

Figure 4b depicts the correlation coefficient  $r_s$  for  $PDEL$  (Y-axis) and the total numbers of errors (X-axis). We see that the SOC utilization, IPC and the DRAM temperature are highly correlated with both metrics as the correlation coefficient  $r_s$  is above 0.6 ( $\rho$ -value is lower than 0.02) for all these parameters which indicates a positive direction of the correlation, i.e. the number of errors and  $PDEL$  grows up with the SOC utilization, IPC and DRAM temperature. We find that IPC and the number of L1 and L2 accesses (both read and write operations) increases with the number of threads and thus the SOC utilization due to a better data locality in parallel applications. Note that the SOC utilization is also highly correlated with the SOC temperature and  $r_s$  is 0.7 ( $\rho$ -value is 0.024) and thus, as we

have shown, the DRAM temperature. Based on these observation, we suggest that the SOC utilization is more correlated with DRAM error behaviour than other parameters as it encapsulates the effect of many system parameters during the program execution and affects the DRAM temperature.

Overall, these results imply that the SOC utilization affects the temperature of DIMMs especially of those that are closer to the SOC. Notably, we also observe a correlation between the SOC utilization and the DRAM temperature on Intel® Xeon based servers, such as [28], which adopt a similar layout with DIMMs being placed adjacent to the SOC on the motherboard. However, the level of this correlation depends on the available cooling system within each server.

### E. Temperature Stress Benchmark

Looking for a suitable method to stress the DRAM temperature which seems to be an important parameter for stressing the DRAMs in the field, we exploited the aforementioned observations and especially the high correlation between the SOC utilization, which is maximized when running multi-threaded workloads, and the number of L2 cache accesses per clock with  $PDEL$  (see Figure 4b).

In particular, we implemented a stress benchmark (SBench), which stresses the SOC utilization by incurring many L2 cache accesses and invoking several parallel threads across all the cores apart from one core where the conventional DPBenchs are being executed for stressing directly the data-dependent circuit level factors on the DRAM. Note that, it was shown that a benchmark with many L2 cache accesses as in [29] when used for energy modelling could significantly increase the SOC power consumption and consequently its temperature. After executing each DPBench in parallel with SBench which spawns 7 threads, we observe the manifestation of 759 CEs in total for both DIMM sets.

Figure 4c shows how the  $PDEL$  averaged over the two DIMM sets changes in time after executing each DPBench with SBench (*random-stress*, *all1s-stress*, *all0s-stress* and *checkerboard-stress*). Each circle and its size represent  $PDEL T(10 \text{ minutes})$  (see Section III). Note that Figure 4c depicts statistics only for error-prone locations discovered with the DPBenchs. It can be observed that the highest  $PDEL$  is achieved for *random-stress*, which is about 60 %, while the other static DPBenchs discover less than 23 % of all the error-prone locations. We also observe that  $PDEL T(10 \text{ minutes})$  drops to 0 for almost all the DPBenchs after 100 minutes of experiments, which implies that 2 hours of experiments should be enough to cover the majority of error-prone locations. These results

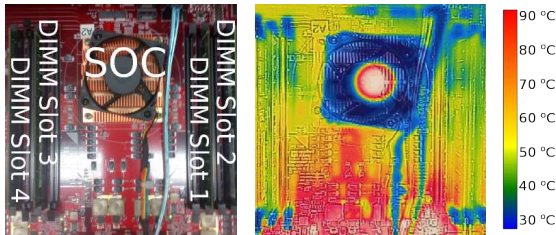


Fig. 5: DRAM slots on X-Gen2 and corresponding thermal image.

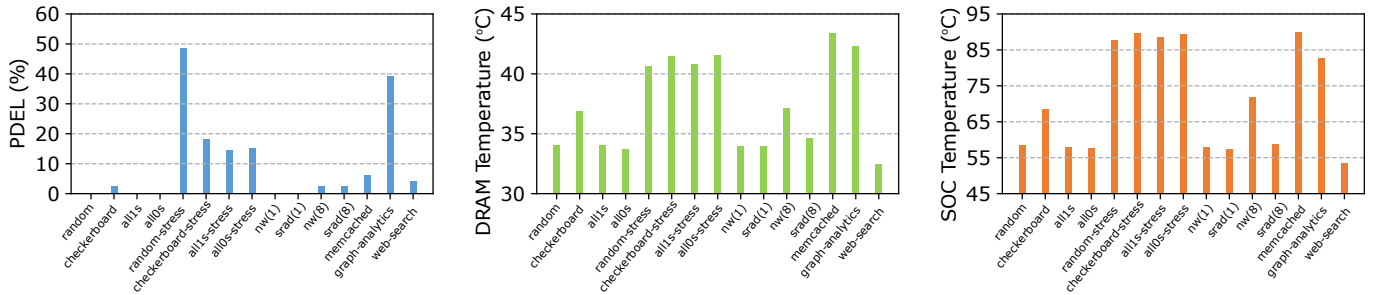


Fig. 6: *PDEL* considering error location of all the workloads, SOC and DRAM temperatures.

are consistent with observations made in [10], where authors also found that the *random* DPBench covers the highest percentage of failing cells.

Figure 6 shows *PDEL*, SOC and DRAM temperatures measured after running 2 hours of each benchmark, including DPBenchs with the co-running SBench, averaged over all DIMM sets. *Random-stress* covers the highest number of unique error-prone locations, but this number is 13 % less than the percentage of discovered error-prone locations observed in Figure 4c. These findings suggest that real applications trigger errors in a few memory locations which are not covered by DPBenchs. Furthermore, DRAM and SOC temperatures measured for the DPBenchs grow above 40 °C and 85 °C correspondingly when running with SBench (see Figure 6). Nonetheless, *random-stress* covers many more error-prone locations than the static DPBenchs. This difference is attributable to specific data and memory access patterns which, as previous research studies shown [9], may significantly affect memory error behaviour.

#### F. Power reduction

To estimate the power reduction, we run all benchmarks at the nominal  $T_{REFP}$  and relaxed  $T_{REFP}$  and take DRAM power measurements using on board DIMM power sensors. By relaxing  $T_{REFP}$  from 64 ms up to 2.283 s, we manage to reduce the total memory power by 11.5 % on average without compromising reliability as all errors were corrected by ECC.

#### V. CONCLUSION

In this paper, we present a comprehensive study on DRAM reliability characterization under relaxed refresh period within a commodity server using conventional data patterns along with a set of HPC and Cloud workloads under non-controlled as well as controlled temperatures. We demonstrate that the excited error-prone locations vary from workload to workload and differ from the ones discovered by the conventional data patterns. Our results suggest that conventional data patterns are not effective for DRAM characterization without any temperature stress mechanism. Such a mechanism was implemented by complicated thermal testbeds which are anyway not going to be available in the field for online characterization. In addition, we quantify for the first time the indirect impact of system level factors, such as of the SOC utilization, on DRAM reliability, which may have been suspected before in data centers but was never investigated. These facts led to the development of a stress benchmark, which raises the temperature when executed in parallel with data patterns and significantly increases the number of discovered error-prone locations, thus facilitating DRAM characterization in the field without any thermal testbed. Finally, we show that the DRAM refresh period can be relaxed by 35x on such a commodity system with all errors being corrected by the available ECC when the DRAM temperature varies from 33 °C up to 43 °C, resulting in 11.5% power savings.

#### REFERENCES

- [1] Cisco Systems, "Cisco global cloud index: Forecast and methodology 2015-2020," 2016.
- [2] H. David *et al.*, "RAPL: Memory power estimation and capping," in *Proceedings of the 16th ACM/IEEE ISLPD*, 2010, pp. 189–194.
- [3] J. Liu *et al.*, "RAIDR: Retention-Aware Intelligent DRAM Refresh," ser. ISCA. Washington, DC, USA: IEEE Computer Society, 2012.
- [4] J. Kim *et al.*, "Block-based multiperiod dynamic memory design for low data-retention power," *VLSI*, vol. 11, no. 6, pp. 1006–1018, Dec 2003.
- [5] T. Ohsawa *et al.*, "Optimizing the dram refresh count for merged dram/logic lsis," ser. ISLPED. NY, USA: ACM, 1998, pp. 82–87.
- [6] H.-H. S. Lee *et al.*, "Smart refresh: An enhanced memory controller design for reducing energy in conventional and 3d die-stacked drams," *MICRO*, vol. 00, pp. 134–145, 2007.
- [7] R. K. Venkatesan *et al.*, "Retention-aware placement in dram (rapid): software methods for quasi-non-volatile dram," in *HPCA*, 2006.
- [8] K. Tovletoglou *et al.*, "Relaxing dram refresh rate through access pattern scheduling: A case study on stencil-based algorithms," in *IOLTS*, 2017.
- [9] S. Khan *et al.*, "The efficacy of error mitigation techniques for dram retention failures: A comparative experimental study," *SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 1, pp. 519–532, Jun. 2014.
- [10] J. Liu *et al.*, "An experimental study of data retention behavior in modern dram devices: Implications for retention time profiling mechanisms," in *ISCA*, NY, USA, 2013, pp. 60–71.
- [11] M. Patel *et al.*, "The reach profiler: Enabling the mitigation of dram retention failures via profiling at aggressive conditions," *ISCA*, 2017.
- [12] M. Jung *et al.*, "A platform to analyze ddr3 dram's power and retention time," *IEEE Design Test*, vol. 34, no. 4, pp. 52–59, Aug 2017.
- [13] B. Schroeder *et al.*, "DRAM Errors in the Wild: A Large-scale Field Study," in *Proceedings of SIGMETRICS 2009*, pp. 193–204.
- [14] N. El-Sayed *et al.*, "Temperature management in data centers: Why some (might) like it hot," ser. SIGMETRICS '12.
- [15] P. J. Nair *et al.*, "Archshield: Architectural framework for assisting dram scaling by tolerating high error rates," *SIGARCH Comput. Archit.*, 2013.
- [16] C.-H. Lin *et al.*, "Secret: A selective error correction framework for refresh energy reduction in drams," *ACM TACO*, Jun. 2015.
- [17] T. Hamamoto *et al.*, "On the retention time distribution of dynamic random access memory (dram)," *IEEE Electron Devices*, 1998.
- [18] Y. H. Son *et al.*, "Reducing memory access latency with asymmetric dram bank organizations," ser. ISCA, 2013.
- [19] Atx specification. [Online]. Available: <http://www.formfactors.org/developer/5Cspecs/5Catx2-2.PDF>
- [20] G. Singh *et al.*, "AppliedMicro X-Gene2," in *Hot Chips*, 2014.
- [21] Micron Technology, "DDR3 - 8GB," 2015. [Online]. Available: <https://www.micron.com/parts/modules/ddr3-sdram/mt18jsf1g72az-1g9>
- [22] Micron Technology, "DDR3 SDRAM MT41J512M8," 2009.
- [23] M. J. Lee *et al.*, "A mechanism for dependence of refresh time on data pattern in dram," *IEEE Electron Device Letters*, 2010.
- [24] S. Che *et al.*, "Rodinia: A benchmark suite for heterogeneous computing," ser. IISWC. USA: IEEE Computer Society, 2009, pp. 44–54.
- [25] Z. Ou *et al.*, "Energy- and cost-efficiency analysis of arm-based clusters," in *CCGRID*, May 2012, pp. 115–123.
- [26] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 1988.
- [27] Micron Technology, "TN-04-42: Memory Module Serial Presence-Detect," 2002.
- [28] Supermicro, "SuperServer 1028GQ-TR." [Online]. Available: [www.supermicro.com/products/system/1u/1028/sys-1028gq-tr.cfm](http://www.supermicro.com/products/system/1u/1028/sys-1028gq-tr.cfm)
- [29] L. Mukhanov *et al.*, "Alea: Fine-grain energy profiling with basic block sampling," in *PACT*, Oct 2015, pp. 87–98.